5 - Protocols

21S1

Q6: What is an idempotent operation? Some of the primitive operations for a typical flat file service interface for a Distributed File System are shown below (UFID stands for Unique File Identifier). Which of the following primitives of the interface are not idempotent? Explain your answer.

(A.) Read(UFID, i, n): Reads up to n items from position i in the file.

Yes

(B.) Write(UFID, i, Data): Writes the data starting at position i in the file. The file is extended if necessary.

Yes

(C.) Create(): Creates a new file of length 0 and returns a UFID for it.

No, will create multiple files.

(D.) Delete(UFID): Removes the file from the file store/system.

Yes.

17S1

Q4: (a) [5 marks] Suppose that a client creates a TCP Socket to a server, sends a command, and waits for a response. However the server has failed internally and does not respond, though the connection remains open. What can be done by the client to overcome this? Critically explain your answer and include discussion concerning relevant distributed system challenges.

A timeout mechanism can be added. For example, the client can send 10 additional requests to the server to test if the server is still alive. If the server still does not respond, then it is highly likely that the server is down and the client can chose to close the connection.

However, the server may have a higher load if implementing this mechanism.

(b) [5 marks] Explain what is meant by a remote procedure call or RPC. Discuss the aspects of implementing an RPC middleware such as Sun RPC. Draw a diagram to show the architectural components of the implementation.

RPCs enable clients to execute procedures in server processes based on a defined service interface. P88

Sun NFS: P130

- It follows the NFS protocol, using a set of RPCs that provide the means for the client to perform operations on the remote file store.

16S2

Q4: (c) Explain maybe semantics, at-most-once semantics and at-least-once semantics, in terms of a client sending a request to a server.

What level of semantics does the RR protocol achieve?

- Request/Reply (RR) protocol -> At-least-once L5 P9

What level of semantics does Java RMI enforce?

- Java RMI provides *at-most-once* invocation semantics only.

Other

Describe 3 invocation semantics: Maybe, At-least-once, At-most-once. What is the difference between Maybe and At-most-once?

Maybe: The client does not know if the request is processed since there is no response form the server.

At-least-once: The client knows that its request has been processed at least once if a response is received. If no response is received, the request is not processed by the server.

At-most-once: The client knows that its request has been processed at most once if a response is received. If no response is received, the request is not processed by the server.

Answer the following questions about Java RMI:

1. **Can objects be created remotely? Explain your answer.**
    - No, object cannot be directly created remotely.
    - It is like a proxy for the remote object which can be operated remotely.

2. **Explain the difference between a local invocation and a remote invocation. If two Java Virtual Machines are on the same physical machine, and invocation is made between them, is this local or remote?**

    - Local invocation takes place between objects within the same JVM.
    - Remote invocation takes place between objects on different JVMs.
    - Though on the same physical machine, this is still a remote invocation.

3. **What is a remote reference? Explain how an object obtains a remote reference.**

    - Remote reference is the object that is not within the local JVM.
    - By accessing the registry process, the remote object can be bind if its name is known.